

ORIGINAL

PATENT APPLICATION

ATTORNEY DOCKET NO. 200304479-1

AF/2176
IFW
#



IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Monika R. HENZINGER et al.

Confirmation No.: 9654

Application No.: 09/392,170

Examiner: A. R. Yuan

Filing Date: 09/08/1999

Group Art Unit: 2176

Title: RANKING SEARCH ENGINE RESULTS

Mail Stop Appeal Brief-Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Sir:

Transmitted herewith in **triplicate** is the Appeal Brief in this application with respect to the Notice of Appeal filed on 06/07/2004.

The fee for filing this Appeal Brief is (37 CFR 1.17(c)) \$330.00.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

() (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d) for the total number of months checked below:

() one month	\$110.00
() two months	\$420.00
() three months	\$950.00
() four months	\$1480.00

() The extension fee has already been filled in this application.

(X) (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account **08-2025** the sum of \$330.00. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees. A duplicate copy of this sheet is enclosed.

(X) I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313-1450. Date of Deposit: 07/12/2004
OR

() I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number _____ on _____

Number of pages:

Typed Name: Christina L. Paz

Signature: Christina L. Paz

Respectfully submitted,

Monika R. HENZINGER et al.

By Jonathan M. Harris

Jonathan M. Harris

Attorney/Agent for Applicant(s)

Reg. No. 44,144

Date: 07/12/2004

Telephone No.: (713) 238-8000



ORIGINAL

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellants:	Monika R. Henzinger et al.	§	Confirmation No.:	9654
		§		
Serial No.:	09/392,170	§	Group Art Unit:	2176
		§		
Filed:	09/08/1999	§	Examiner:	A. R. Yuan
		§		
For:	Ranking Search	§	Docket No.:	200304479-1
	Engine Results	§		

APPEAL BRIEF

Mail Stop Appeal Brief – Patents

Date: July 12, 2004

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

Appellants hereby submit this Appeal Brief in connection with the above-identified application. A Notice of Appeal was filed on June 7, 2004.

I. REAL PARTY IN INTEREST

The real party in interest is the Hewlett-Packard Development Company (HPDC), a Texas Limited Partnership, having its principal place of business in Houston, Texas, through its merger with Compaq Computer Corporation (CCC) which owned Compaq Information Technologies Group, L.P. (CITG). The assignment from the CCC to CITG was recorded on 12/28/2001, at Reel/Frame 012403/0178. The Change of Name document (CITG to HPDC) was recorded on 05/12/2004, at Reel/Frame 014628/0103.

II. RELATED APPEALS AND INTERFERENCES

Appellants are unaware of any related appeals or interferences.

III. STATUS OF THE CLAIMS

Originally filed claims: 1-59.
Claim cancellations: 1, 29, 33, 34 and 43.
Added claims: 60-62.
Presently pending claims: 2-28, 30-32, 35-42 and 44-62.
Presently appealed claims: 2-28, 30-32, 35-42 and 44-62.

07/16/2004 WABDELRI 00000087 082025 09392170
01 FC:1402 330.00 DA

IV. STATUS OF THE AMENDMENTS

No claims were amended after the final Office action dated April 20, 2004.

V. SUMMARY OF THE INVENTION

The search engines and their associated search indices available on the World Wide Web are of varying degrees of quality where quality is defined as being "how likely they are to return a result that will be of use to the user." Page 4, lines 20-21. Traditionally, this qualitative assessment has been based on the relative sizes of the indices, *i.e.*, the number pages indexed. However, a qualitative assessment based on size alone does not take into account the relative quality of the pages that are indexed. The relative quality of the pages retrieved in a web search may be of greater importance than how many pages are retrieved. See page 4, line 24 – page 5, line 9.

Appellants' contributions provide systems and methods for objectively measuring and ranking search engine indices and/or search engine results based on the relative quality of the included pages. See page 5, lines 21-23. Embodiments disclosed in the application employ a two-level random walk to generate a measure of page quality. See *e.g.*, page 6, lines 3-4. In general, a two-level random walk starts with an initial host and random selection of a page in that host. Then, a random decision is made as to whether to follow a link on the selected page or to randomly select a new page. If the decision is to follow a link on the selected page, a link is randomly selected on that page. If the decision is to randomly select a new page, a host is randomly selected from the hosts encountered thus far in the walk and a page is then randomly selected from the chosen host. This process continues until some predetermined condition is met or until all pages have been traversed. See page 13, line 24 – page 14, line 11.

The percentage of documents in a search engine index that are found during the two-level random walk can be used to determine the quality of that search engine index. The quality of a particular document can be determined based on how many times the document is found during the two-level random walk. See page 6, lines 15-20.

VI. ISSUES

The issues in this Appeal are: 1) whether claims 2, 4-9, 11-12, 15-19, 24-25, 27-28, 30, 32, 35-37, 44-47, 52-57 and 60-62 are patentable under 35 U.S.C. § 103(a) over Pitkow et al. (U.S. Pat. No. 6,457,028) in view of Singhal (U.S. Pat. No. 6,370,527); 2) whether claims 3, 10, 31 and 38 are patentable under § 103(a) over Pitkow and Singhal and further in view of Page (U.S. Pat. No. 6,285,999); 3) whether claims 13-14, 20-22, 26, 41-42, 48-50, 54 and 58-59 are patentable under § 103(a) over Pitkow; and 4) whether claims 23 and 51 are patentable under § 103(a) over Pitkow in view of Page.

VII. GROUPING OF CLAIMS

Appellants propose the following claim groupings with the number in parentheses referring to a representative claim in each group:

- (1) claims 2-6, 27, 30-32, 55, 57 (1);
- (2) claims 7, 8, 11, 12, 35, 36, 39, 40 (7);
- (3) claims 9, 10, 37, 38 (9);
- (4) claims 13, 14, 16-19, 41, 42, 44-47, 58 (13);
- (5) claim 15 (15);
- (6) claims 20, 48, 59 (20);
- (7) claims 21-23, 26, 49-51, 54 (21);
- (8) claims 24, 52 (24);
- (9) claims 25, 28, 53, 56 (25);
- (10) claims 60-62 (60).

The claims in each group are patentable separately from the claims of the other groups and, as such, do not stand or fall with the claims of the other groups as argued below.

VIII. ARGUMENT

A. The Pitkow Reference

Pitkow discloses a system and method for analyzing collections of web pages and/or web sites to identify collections having similar content. See Pitkow at col. 1 lines 15-19, col. 3 lines 1-15. Citation analysis techniques and clustering operations are applied to a collection of web pages/sites to identify clusters of

closely related pages and/or sites in the collection. See Pitkow at col. 3 lines 3-15. The first step of this analysis is to obtain a collection of documents (e.g., web pages). See Pitkow at col.1 lines 20-23, col. 7 lines 48-49, col. 10 lines 17-18. Several options for obtaining this initial collection of documents are suggested such as a specific Web query using some type of document retrieval system, a Web walker, a randomly selected collection of documents, a particular Web site or set of web sites, or the entire Web. See Pitkow at col. 7 lines 49-56, col.10 lines 18-21.

B. The Singhal Reference

Singhal discloses a method and apparatus for searching the World Wide Web using multiple search engines. A meta-search engine receives a query from a user and submits the query to some number of search engines. The results from each search engine are ranked and sorted based on various criteria and then displayed to the user. See Singhal at col. 1 lines 31-41.

C. The Page Reference

Page discloses a method for ranking linked documents on the World Wide Web. The ranking for a document is calculated based on the ranks of documents citing that particular document and a constant representing the probability that a user browsing the Web will randomly jump to that document. See Page at Abstract.

Page is used only in the rejection of certain dependent claims which are not argued below. Furthermore, Page does not teach or suggest any of the features that the Examiner attributes to the other art of record.

D. Claim 2

Claim 2 is directed to a computer-implemented method for "randomly walking through a hyper-text-linked document set" that includes, among other features, "d) responsive to the occurrence of a random event: d.1) selecting at random a host from among the previously selected hosts; ... e) responsive to the non-occurrence of the random event: e.1) selecting at random a link in the retrieved document; and e.2) retrieving the document referenced by the selected link; and f) repeating d) and e) until a predetermined condition is met."

The Examiner contends that Pitkow discloses all of the features of the method of claim 2 except "d.1) selecting at random a host from among the previously selected hosts." Appellants disagree with the Examiner. First, Pitkow does not teach or suggest "e.1) selecting at random a link in the retrieved document." Instead, Pitkow merely suggests that a collection of web documents could be randomly selected in some fashion:

[The obtaining of an initial document collection] may be done by a specific query to the Web, using a document retrieval system such as Lycos™, Excite™, etc. or using a Web walker which automatically follows the links on a document and collects the linked documents. **Or, the document collection could be some randomly selected collection of documents.** Or the document collection could be a particular Web site or set of Web sites or even the entire Web itself.

Pitkow, col. 7 lines 49-56 (emphasis added).

The Examiner contends that the above quoted portion of Pitkow teaches "e.1) selecting at random a link in the retrieved document," interpreting this quoted passage to teach:

a web walker that automatically follows links on a document and collects the linked documents; the document collection could be a randomly selected collection of documents, **in other words, the web walker can check the links on the retrieved document and can randomly select and collect the linked documents of the retrieved documents.**

Final Office Action, page 24 (emphasis added).

The Examiner is misconstruing what Pitkow discloses. Pitkow clearly discloses that either an initial document set is obtained by using a web walker that automatically, but not randomly, follows links in a web document, or an initial document set is obtained by some undisclosed form of random selection. Pitkow does not does not teach, suggest, or even imply a web walker that randomly selects and collects linked documents as the Examiner contends. Furthermore, one of ordinary skill in the art at the time of invention would have known that a web crawler or web walker as taught by Pitkow would start with a web page and work systematically and methodically, not randomly, through a process of following all links on that page, following all links on those pages, and so on.

See, e.g., "Frequently Asked Questions about the Mercator Web Crawler," COMPAQ, 1999, available at <http://research.compaq.com/SRC/mercator/faq.html>, and "Writing a Web Crawler in the Java Programming Language," Blum, Thom, *et. al.*, January 1998, available at <http://java.sun.com/developer/technicalArticles/ThirdParty/WebCrawler/> (attached).

Second, Pitkow does not teach or suggest "e) responsive to the non-occurrence of the random event: e.1) selecting at random a link in the retrieved document." At most, Pitkow suggests "automatically," not randomly, following all the links on a document (see Pitkow, col. 7 lines 49-53) or gathering documents "through a web crawl" where linked pages are obtained." Pitkow, col. 10 lines 18-20.

Third, Pitkow does not teach or suggest "f) repeating d) and e) until a predetermined condition is met." The Examiner contends that Pitkow teaches collecting or gathering linked pages "until a list of web sites along with indicators of corresponding web pages are obtained" based on the following portion of Pitkow:

First, a collection of web pages is obtained This collection may be gathered based on a sampling of web pages, through a "web crawl" where linked pages are obtained, or through a query to one or more search engines. In any event, the collection of web pages will be related in some manner. Next, web sites (document collections) are identified based on the web pages Web sites are identified based on the Universal Resource Locator (URL) address of a web page. ... [A] list of web sites along with indicators of the corresponding web page(s) is thus obtained.

Pitkow, col. 10 lines 17-31 (emphasis added).

As can be seen, Pitkow discloses first obtaining a collection of web pages and subsequently analyzing that collection of web pages to determine web sites. This is clearly not collecting or gathering linked pages "until a list of web sites along with indicators of corresponding web pages are obtained" as the Examiner contends, much less "f) repeating d) and e) until a predetermined condition is met."

The Examiner correctly concedes that Pitkow does not disclose "d.1) selecting at random a host from among the previously selected hosts" as

recited by claim 2. See Final Office Action, page 3. The Examiner, however, incorrectly contends that Singhal provides this missing feature. The portion of Singhal relied on by the Examiner discloses that a user of the meta-search engine taught by Singhal may choose a subset of search engine devices to be used for a query rather than having the query go to all search engine devices in order to "limit their search to particular portions of the network." Singhal, col. 7 lines 21-26. User selection from among a subset of search engine devices is not "d.1) selecting at random a host from among the previously selected hosts."

The Examiner has also improperly combined Pitkow and Singhal. The Examiner contends that it would have been obvious to one of ordinary skill in the art to have incorporated the disclosure of Singhal discussed above, *i.e.*, user selection of a subset of search engine devices, into the "web crawl" or one or more search engines taught by Pitkow "in order to allow a user to search all of the available portions of a distributed network without having to repeatedly [sic] reenter their search query." Final Office Action, page 4. Appellants fail to understand the Examiner's logic. The teaching of Singhal that the Examiner relies on clearly teaches that one or more search engine devices are selected by the user to limit the search to particular portions of the network. Singhal, col. 7 lines 21-26. The motivation to combine this feature with Pitkow is stated as being one of allowing a search of all available portions of a network. The Examiner's logic is flawed because one of ordinary skill in the art would not have been led to incorporate a feature of Singhal that **limits a search to particular portions** of a network into the web crawler or one or more search engines disclosed by Pitkow in order to enable **a search of all portions** of a network.

Pitkow and Singhal do not teach or suggest the features that the Examiner attributes to the other. For any or all of the foregoing reasons, the Examiner erred in rejecting claim 2 and its dependent claims over the combination of Pitkow and Singhal.

E. Claim 7

Claim 7 is directed to a computer-implemented method for "randomly walking through a hypertext-linked document set" that includes, among other

features, "c) selecting at random a host from the host set." The method further includes "e) responsive to the selected document containing at least one link: e.1) selecting at random a link from the selected document ... e.6) repeating e.1) through e.5) until a first predetermined condition is met; and f) repeating c) through e) until a second predetermined condition is met."

The Examiner contends that Pitkow teaches many of the features of the method of claim 7. See Final Office Action, page 5-6. For the same or similar reasons as provided in reference to claim 2, Appellants submit that Pitkow does not teach or suggest "e.1) selecting at random a link from the selected document. Singhal does not teach or suggest the features that the Examiner attributes to Pitkow.

The Examiner did not provide any explicit basis for rejecting "e.6) repeating e.1) through e.5) until a first predetermined condition is met" or "f) repeating c) through e) until a second predetermined condition is met." Pitkow and Singhal, alone or in combination, do not teach or suggest either e.6) or f).

The Examiner correctly concedes that Pitkow does not disclose a number of the features included in claim 7 including "c) selecting at random a host from the host set." See Final Office Action, page 6. The Examiner, however, incorrectly contends that Singhal discloses this feature. As explained in reference to claim 2, Singhal teaches user selection of a subset of search engine devices from a collection of search engine devices. This user selection is not "c) selecting at random a host from the host set."

The Examiner has also improperly combined Pitkow and Singhal. As previously explained in reference to claim 2, the Examiner's cited motivation for incorporating the purported teachings of Singhal into Pitkow is logically flawed. One of ordinary skill in the art at the time of the invention would not have incorporated the search limiting feature the Examiner attributes to Singhal into the teachings the Examiner attributes to Pitkow in order to allow searching of all available portions of a network.

For any or all of these reasons, the Examiner erred in rejecting claim 7 and its dependent claims over the combination of Pitkow and Singhal.

F. Claim 9

Claim 9 is directed to a computer-implemented method “for randomly walking through a hypertext-linked document set” that includes, among other features, “c) selecting at random a host from the host set.” The method further includes “e) responsive to non-occurrence of a random event ... : e.1) selecting at random, a link from the selected document; e.2) selecting a document corresponding to the selected link ... e.6) repeating e.1) through e.5) until a first predetermined condition is met.” In addition, the method claims “f) repeating c) through e) until a second predetermined condition is met.”

The Examiner contends that Pitkow teaches many of the features of the method of claim 9. See Final Office Action, pages 7-8. For the same or similar reasons as provided in reference to claim 2, Appellants submit that Pitkow does not teach or suggest “e.1) selecting at random a link from the selected document” or “e) responsive to non-occurrence of a random event ... : e.1) selecting at random, a link from the selected document.” Singhal does not teach or suggest the features that the Examiner attributes to Pitkow.

The Examiner did not provide any explicit basis for rejecting “e.6) repeating e.1) through e.5) until a first predetermined condition is met” or “f) repeating c) through e) until a second predetermined condition is met.” Pitkow and Singhal, alone or in combination, do not teach or suggest either e.6) or f).

The Examiner correctly concedes that Pitkow does not disclose a number of the features included in claim 9 including “c) selecting at random a host from the host set.” See Final Office Action, page 8. The Examiner, however, incorrectly contends that Singhal discloses this feature. As explained in reference to claim 2, Singhal teaches user selection of a subset of search engine devices from a collection of search engine devices. This user selection is not “c) selecting at random a host from the host set.”

The Examiner has also improperly combined Pitkow and Singhal. As previously explained in reference to claim 2, the Examiner’s cited motivation for incorporating the purported teachings of Singhal into Pitkow is logically flawed. One of ordinary skill in the art at the time of the invention would not have

incorporated the search limiting feature the Examiner attributes to Singhal into the teachings the Examiner attributes to Pitkow in order to allow searching of all available portions of a network.

For any or all of these reasons, the Examiner erred in rejecting claim 9 and its dependent claims over the combination of Pitkow and Singhal.

G. Claim 13

Claim 13 is directed to a computer-implemented method for "measuring relative quality of a search engine index" that comprises "a) performing a two-level random walk among documents within a document set; b) for each document encountered in the random walk, determining whether the document is indexed by the search engine index; and c) aggregating the results of b)."

Pitkow does not teach or suggest "a) performing a two-level random walk among documents within a document set." Pitkow discloses obtaining a collection of web documents or pages using one of several alternative approaches followed by performing further processing on that collection. See Pitkow, col. 7 lines 48-59 and col. 10, lines 17-31. The mere obtaining of collection of web documents or pages followed by further processing on that collection, even if the collection is a "randomly selected collection" or is acquired by using a Web walker, is not "performing a two-level random walk among documents within a document set."

Pitkow also does not teach or suggest "b) for each document encountered in the random walk, determining whether the document is indexed by the search engine index." Pitkow discloses "constructing and analyzing a citation index for each web page [in a collection of web pages]. ... [A] citation index is merely a listing of all the links contained in the page." Pitkow, col. 10 lines 56-57. Clearly, a citation index is not a search engine index as required by claim 13 nor is constructing and analyzing a citation index "determining whether the document is indexed by the search engine index." For these same reasons, Pitkow does not teach or suggest "c) aggregating the results of b)."

For any or all of these reasons, the Examiner erred in rejecting claim 13 and its dependent claims over Pitkow.

H. Claim 15

Claim 15 is directed to a computer-implemented method for "measuring relative quality of a search engine index" that includes, among other features, "a) performing a two-level random walk among documents within a document set, by: ... a.3.1) responsive to occurrence of a random event: a.3.1.1) selecting at random a host from among the previously selected hosts." The two-level random walk further includes "a.3.2) responsive to non-occurrence of the random event: a.4) selecting at random a link in the retrieved document; ... and a.6) repeating a.3.1) through a.5) until a predetermined condition is met." The method also includes "b) for each document encountered in the random walk, determining whether the document is indexed by the search engine index; and c) aggregating the results of b)."

For the same or similar reasons as those given in reference to claim 2, the combination of Pitkow and Singhal does not teach or suggest "a.3.1.1) selecting at random a host from among the previously selected hosts," "a.4) selecting at random a link in the retrieved document," "a.3.1) responsive to occurrence of a random event: a.3.1.1) selecting at random a host from among the previously selected hosts," or "a.3.2) responsive to non-occurrence of the random event: a.4) selecting at random a link in the retrieved document."

The Examiner provided no explicit basis for rejecting "a.6) repeating a.3.1) through a.5) until a predetermined condition is met." Pitkow and Singhal, alone or in combination, do not teach this limitation of claim 15.

Furthermore, as explained in reference to claim 13, Pitkow does not teach or suggest either "b) for each document encountered in the random walk, determining whether the document is indexed by the search engine index" or "c) aggregating the results of b)."

The Examiner has also improperly combined Pitkow and Singhal. As previously explained in reference to claim 2, the Examiner's cited motivation for incorporating the purported teachings of Singhal into Pitkow is logically flawed. One of ordinary skill in the art at the time of the invention would not have incorporated the search limiting feature the Examiner attributes to Singhal into the

teachings the Examiner attributes to Pitkow in order to allow searching of all available portions of a network.

For any or all of these reasons, the Examiner erred in rejecting claim 15 over the combination of Pitkow and Singhal.

I. Claim 20

Claim 20 is directed to a computer-implemented method “for measuring relative quality of a target document in a document set” that includes “a) performing a two-level random walk among documents within a document set; and b) determining a quality metric responsive to the number of times the target document is encountered in the random walk.”

As explained above in reference to claim 13, Pitkow does not teach or suggest “a) performing a two-level random walk among documents within a document set.”

Pitkow does not teach or suggest “b) determining a quality metric responsive to the number of times the target document is encountered in the random walk.” Pitkow discloses analyzing a collection of web pages obtained by using one of several listed alternatives to determine co-citation pairs. See Pitkow, col. 10 lines 15-52. The analysis to determine co-citation pairs includes determining “the number of times each of the sites are both cited by the same page and thus the same site.” Pitkow, col. 10 lines 56-59. These co-citation pairs are then analyzed “to identify collections of related web sites.” Pitkow, col. 10 lines 61-63. This determination of co-citation pairs is not “b) determining a quality metric responsive to the number of times the target document is encountered in the random walk.”

For any or all of these reasons, the Examiner erred in rejecting claim 20 over Pitkow.

J. Claim 21

Claim 21 is directed to a computer-implemented method “for measuring relative quality of a target document in a document set” that includes “a) performing a two-level random walk among documents within a document set;

and b) determining a quality metric responsive to the number of documents encountered during the two-level random walk that link to the target document.”

As previously explained in reference to claim 13, Pitkow does not teach or suggest includes “a) performing a two-level random walk among documents within a document set.” Furthermore, for the same or similar reasons as presented in reference to claim 20 above, Pitkow does not teach or suggest “b) determining a quality metric responsive to the number of documents encountered during the two-level random walk that link to the target document.”

For any or all of these reasons, the Examiner erred in rejecting claim 21 and its dependent claims over Pitkow.

K. Claim 24

Claim 24 is directed to a computer-implemented method “for measuring relative of a target document in a document set” that includes, among other features, “a) performing a two-level random walk among documents within a document set by: ... a.4) responsive to occurrence of a random event: a.4.1) selecting at random a host from among the previously selected hosts.” The two-level random walk further includes: “a.5) responsive to non-occurrence of the random event: a.5.1) selecting at random a link in the retrieved document; ... and a.6) repeating a.4) to a.5) until a predetermined condition is met.” The method further includes “b) determining a quality metric responsive to the number of documents encountered during the two-level random walk that link to the target document.”

For the same or similar reasons as given in reference to claim 2, the combination of Pitkow and Singhal does not teach or suggest the detailed two-level random walk as recited in claim 24. Specifically, Pitkow and Singhal do not teach or suggest “a.5.1) selecting at random a link in the retrieved document” or “a.4.1) selecting at random a host from among the previously selected hosts.” In addition, “a.4) responsive to occurrence of a random event: a.4.1) selecting at random a host from among the previously selected hosts” and “a.5) responsive to non-occurrence of the random event: a.5.1) selecting at random a link in the

retrieved document” are not taught or suggested by the combination of Pitkow and Singhal.

The Examiner provided no explicit basis for rejecting “a.6) repeating a.4) to a.5) until a predetermined condition is met.” Pitkow combined with Singhal does not teach or suggest this limitation of claim 24.

Furthermore, for the same or similar reasons as provided in reference to claim 20 above, Pitkow does not teach or suggest “b) determining a quality metric responsive to the number of documents encountered during the two-level random walk that link to the target document.”

The Examiner has also improperly combined Pitkow and Singhal. As previously explained in reference to claim 2, the Examiner’s cited motivation for incorporating the purported teachings of Singhal into Pitkow is logically flawed. One of ordinary skill in the art at the time of the invention would not have incorporated the search limiting feature the Examiner attributes to Singhal into the teachings the Examiner attributes to Pitkow in order to allow searching of all available portions of a network.

For any or all of these reasons, the Examiner erred in rejecting claim 24 over the combination of Pitkow and Singhal.

L. Claim 25

Claim 25 is directed to a computer-implemented method for “measuring relative of a target document in a document set” that includes, among other features, “a) performing a two-level random walk among documents within a document set, by: ... a.3) selecting at random a host from the host set; a.4) responsive to occurrence of a random event: a.4.1) selecting at random a host from among the previously selected hosts.” The two-level random walk further includes: “a.5) responsive to non-occurrence of the random event: ... a.5.2.1) selecting at random a link from the selected document.” In addition, the random walk includes “a.5.2.6) repeating a.5.2.1) through a.5.2.5) until a first predetermined condition is met; and a.6) repeating a.3 through a.5) until a second predetermined condition is met.” The method further includes “b) determining a

quality metric responsive to the number of documents encountered during the two-level random walk that link to the target document.”

For the same or similar reasons as given in reference to claim 2, the combination of Pitkow and Singhal do not teach or suggest the “two-level random walk” as recited in claim 25. Specifically, Pitkow and Singhal do not teach or suggest “a.3) selecting at random a host from the host set,” “a.4.1) selecting at random a host from among the previously selected hosts,” or “a.5.2.1) selecting at random a link from the selected document.” Further, “a.4) responsive to occurrence of a random event: a.4.1) selecting at random a host from among the previously selected hosts,” and “a.5) responsive to non-occurrence of the random event: ... a.5.2.1) selecting at random a link from the selected document” are not taught or suggested by the combination of Pitkow and Singhal.

The Examiner provided no explicit basis for rejecting “a.5.2.6) repeating a.5.2.1) through a.5.2.5) until a first predetermined condition is met,” or “a.6) repeating a.3 through a.5) until a second predetermined condition is met.” Pitkow and Singhal, alone or in combination, do not teach or suggest this limitation of claim 25.

Furthermore, for the same or similar reasons as provided in reference to claim 20 above, Pitkow does not teach or suggest “b) determining a quality metric responsive to the number of documents encountered during the two-level random walk that link to the target document.”

The Examiner has also improperly combined Pitkow and Singhal. As previously explained in reference to claim 2, the Examiner’s cited motivation for incorporating the purported teachings of Singhal into Pitkow is logically flawed. One of ordinary skill in the art at the time of the invention would not have incorporated the search limiting feature the Examiner attributes to Singhal into the teachings the Examiner attributes to Pitkow in order to allow searching of all available portions of a network.

For any or all of these reasons, the Examiner erred in rejecting claim 25 and its dependent claims over the combination of Pitkow and Singhal.

M. Claim 60

Claim 60 is directed to a system that includes, among other features, "a processor ... wherein the processor initializes a document set, selects an arbitrary hyperlink included in a selected document in the document set, and adds a document referenced by the hyperlink to the document set."

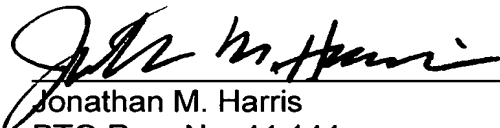
The Examiner apparently contends that Pitkow discloses selecting "an arbitrary hyperlink" because Pitkow discloses using a web walker that automatically follows links on a document. See Final Office Action, Page 17. Automatically following links is not selecting "an arbitrary hyperlink" as required by Claim 60. Singhal does not satisfy this deficiency of Pitkow.

For any or all of these reasons, the Examiner erred in rejecting claim 60 and its dependent claims over the combination of Pitkow and Singhal.

IX. CONCLUSION

For the reasons stated above, Appellants respectfully submit that the Examiner erred in rejecting all pending claims. If any fees or time extensions are inadvertently omitted or if any fees have been overpaid, please appropriately charge or credit those fees to Hewlett-Packard Company Deposit Account Number 08-2025 and enter any time extension(s) necessary to prevent this case from being abandoned.

Respectfully submitted,



Jonathan M. Harris
PTO Reg. No. 44,144
CONLEY ROSE, P.C.
(713) 238-8000 (Phone)
(713) 238-8008 (Fax)
ATTORNEY FOR APPELLANTS

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
Legal Dept., M/S 35
P.O. Box 272400
Fort Collins, CO 80527-2400

APPENDIX "A" TO APPEAL BRIEF
CURRENT CLAIMS

1. (Cancelled).
2. (Previously presented) A computer-implemented method for randomly walking through a hyper-text-linked document set comprising a plurality of documents, wherein at least a subset of the documents contain a plurality of links to other documents, each document being associated with a host, the method comprising:
 - a) selecting a host;
 - b) selecting at random a document associated with the host;
 - c) retrieving the selected document;
 - d) responsive to occurrence of a random event:
 - d.1) selecting at random a host from among the previously selected hosts;
 - d.2) selecting at random a document associated with the host;
 - and
 - d.3) retrieving the selected document;
 - e) responsive to non-occurrence of the random event:
 - e.1) selecting at random a link in the retrieved document; and
 - e.2) retrieving a document referenced by the selected link; and
 - f) repeating d) and e) until a predetermined condition is met.
3. (Previously presented) The method of claim 2, wherein the random event comprises:
 - a generated random number falling within a predetermined range.
4. (Previously presented) The method of claim 2, wherein the document set is the World Wide Web, and wherein each document is a web page.

5. (Original) The method of claim 4, wherein each host corresponds to a domain.

6. (Previously presented) The method of claim 2, further comprising, concurrently with a) through f), performing a second two-level random walk through the hypertext-linked document set.

7. (Previously presented) A computer-implemented method for randomly walking through a hypertext-linked document set comprising a plurality of documents, wherein at least a subset of the documents contain a plurality of links to other documents, each document being associated with a host, the method comprising:

- a) initializing a host set;
- b) initializing a document set for each host in the host set;
- c) selecting at random a host from the host set;
- d) selecting at random a document from the document set of the selected host;
- e) responsive to the selected document containing at least one link:
 - e.1) selecting at random a link from the selected document;
 - e.2) selecting a document corresponding to the selected link;
 - e.3) selecting a host corresponding to the selected document;
 - e.4) adding the selected host to the host set;
 - e.5) adding the selected document to the document set of the selected host; and
 - e.6) repeating e.1) through e.5) until a first predetermined condition is met; and
- f) repeating c) through e) until a second predetermined condition is met.

8. (Previously presented) The method of claim 7, wherein:
- e.4) is performed responsive to the selected host not being in the host set; and
 - e.5) is performed responsive to the selected document not being in the document set of the selected host.
9. (Previously presented) A computer-implemented method for randomly walking through a hypertext-linked document set comprising a plurality of documents, wherein at least a subset of the documents contain a plurality of links to other documents, each document being associated with a host, the method comprising:
- a) initializing a host set;
 - b) initializing a document set for each host in the host set;
 - c) selecting at random a host from the host set;
 - d) selecting at random a document from the document set of the selected host;
 - e) responsive to non-occurrence of a random event and further responsive to the selected document containing at least one link:
 - e.1) selecting at random a link from the selected document;
 - e.2) selecting a document corresponding to the selected link;
 - e.3) selecting a host corresponding to the selected document;
 - e.4) adding the selected host to the host set;
 - e.5) adding the selected document to the document set of the selected host; and
 - e.6) repeating e.1) through e.5) until a first predetermined condition is met; and
 - f) repeating c) through e) until a second predetermined condition is met.

10. (Previously presented) The method of claim 9, wherein the random event comprises:

a generated random number falling within a predetermined range.

11. (Original) The method of claim 7, wherein the hypertext-linked document set is the World Wide Web, and wherein each document is a web page.

12. (Original) The method of claim 11, wherein each host corresponds to a domain.

13. (Original) A computer-implemented method for measuring relative quality of a search engine index, comprising:

- a) performing a two-level random walk among documents within a document set;
- b) for each document encountered in the random walk, determining whether the document is indexed by the search engine index; and
- c) aggregating the results of b).

14. (Original) The method of claim 13, wherein at least a subset of the documents contain a plurality of links to other documents, each document being associated with a host, and wherein a) comprises:

- a.1) selecting a host;
- a.2) selecting at random a document associated with the host;
- a.3) retrieving the selected document;
- a.4) selecting at random a link in the retrieved document;
- a.5) retrieving a document referenced by the selected link; and
- a.6) repeating a.4) and a.5) until a predetermined condition is met.

15. (Previously presented) A computer-implemented method for measuring relative quality of a search engine index, comprising:

- a) performing a two-level random walk among documents within a document set, by:
 - a.1) selecting a host;
 - a.2) selecting at random a document associated with the host;
 - a.3) retrieving the selected document;
 - a.3.1) responsive to occurrence of a random event;
 - a.3.1.1) selecting at random a host from among the previously selected hosts;
 - a.3.1.2) selecting at random a document associated with the host; and
 - a.3.1.3) retrieving the selected document;
 - a.3.2) responsive to non-occurrence of the random event:
 - a.4) selecting at random a link in the retrieved document; and
 - a.5) retrieving a document referenced by the selected link; and
 - a.6) repeating a.3.1) through a.5) until a predetermined condition is met;
- b) for each document encountered in the random walk, determining whether the document is indexed by the search engine index; and
- c) aggregating the results of b).

16. (Previously presented) The method of claim 13, wherein at least a subset of the documents contain a plurality of links to other documents, each document being associated with a host, and wherein a) comprises:

- a.1) initializing a host set;
- a.2) initializing a document set for each host in the host set;
- a.3) selecting at random a host from the host set;
- a.4) selecting at random a document from the document set of the selected host;
- a.5) adding the a host that is referenced by the selected link to the host set;

- a.6) adding the document referenced by the selected link to the document set of the selected host;
 - a.7) responsive to the selected document containing at least one link:
 - a.7.1) selecting at random a link from the selected document;
 - a.7.2) selecting a document corresponding to the selected link;
 - a.7.3) selecting a host corresponding to the selected document;
 - a.7.4) repeating a.5) through a.8) until a predetermined condition is met; and
 - a.8) responsive to the selected document not containing at least one link, repeating a.3) through a.8) until a predetermined condition is met.
17. (Original) The method of claim 16, wherein:
- a.5) is performed responsive to the selected host not being in the host set; and
 - a.6) is performed responsive to the selected document not being in the document set of the selected host.
18. (Original) The method of claim 13, wherein each document contains a plurality of words, and wherein b) comprises, for each document encountered in the random walk:
- b.1) selecting at least one word from the document;
 - b.2) performing a query on the search engine index based on the selected at least one word, to obtain search results; and
 - b.3) determining whether the document is included in the obtained search results.
19. (Original) The method of claim 18, wherein b.1) comprises selecting at least one word based on rarity.

20. (Previously presented) A computer-implemented method for measuring relative quality of a target document in a document set, comprising:

- a) performing a two-level random walk among documents within a document set; and
- b) determining a quality metric responsive to the number of times the target document is encountered in the random walk.

21. (Previously presented) A computer-implemented method for measuring relative quality of a target document in a document set comprising a plurality of documents, wherein at least a subset of the documents contain a plurality of links to other documents, the method comprising:

- a) performing a two-level random walk among documents within a document set; and
- b) determining a quality metric responsive to the number of documents encountered during the two-level random walk that link to the target document.

22. (Previously presented) The method of claim 21, wherein b) comprises determining a quality metric responsive to the number of documents that link to the target document, and responsive to the quality metric of the linking documents.

23. (Previously presented) The method of claim 21, wherein b) comprises determining a value for:

$$R(p) = d/T + (1-d) \sum_{i=1}^k R(p_i)/C(p_i)$$

where:

R(p) is the PageRank of target document p;

R(p_i) is the PageRank of document p_i;

T is the total number of documents in the document set;

d is a damping factor such that 0 < d < 1;

documents p_i, \dots, P_k each contain at least one link to target document p ;
and

$C(p_i)$ is the number of links out of document p_i .

24. (Previously presented) A computer-implemented method for measuring relative quality of a target document in a document set comprising a plurality of documents, wherein at least a subset of the documents contain a plurality of links to other documents, wherein each document is associated with a host, the method comprising:

- a) performing a two-level random walk among documents within a document set by:
 - a.1) selecting a host;
 - a.2) selecting at random a document associated with the host;
 - a.3) retrieving the selected document;
 - a.4) responsive to occurrence of a random event:
 - a.4.1) selecting at random a host from among the previously selected hosts;
 - a.4.2) selecting at random a document associated with the host; and
 - a.4.3) retrieving the selected document;
 - a.5) responsive to non-occurrence of the random event:
 - a.5.1) selecting at random a link in the retrieved document; and
 - a.5.2) retrieving a document referenced by the selected link; and
 - a.6) repeating a.4) to a.5) until a predetermined condition is met; and
- b) determining a quality metric responsive to the number of documents encountered during the two-level random walk that link to the target document.

25. (Previously presented) A computer-implemented method for measuring relative quality of a target document in a document set comprising a plurality of documents, wherein at least a subset of the documents contain a plurality of links to other documents, wherein each document is associated with a host, the method comprising:

- a) performing a two-level random walk among documents within a document set, by:
 - a.1) initializing a host set;
 - a.2) initializing a document set for each host in the host set;
 - a.3) selecting at random a host from the host set;
 - a.4) responsive to occurrence of a random event:
 - a.4.1) selecting at random a host from among the previously selected hosts;
 - a.5) responsive to non-occurrence of the random event:
 - a.5.1) selecting at random a document from the document set of the selected host; and
 - a.5.2) responsive to the selected document containing at least one link:
 - a.5.2.1) selecting at random a link from the selected document;
 - a.5.2.2) selecting a document corresponding to the selected link;
 - a.5.2.3) selecting a host corresponding to the selected document; and
 - a.5.2.4) adding the selected host to the host set;
 - a.5.2.5) adding the selected document to the document set of the selected host;
 - a.5.2.6) repeating a.5.2.1) through a.5.2.5) until a first predetermined condition is met; and

- a.6) repeating a.3) through a.5) until a second predetermined condition is met; and
 - b) determining a quality metric responsive to the number of documents encountered during the two-level random walk that link to the target document.
26. (Previously presented) The method of claim 21, further comprising:
- c) determining a quality metric for at least one additional target document; and
 - d) ranking the quality metric of the first target document with respect to the quality metrics of the additional target documents.
27. (Previously presented) A computer-implemented method for randomly walking through a hypertext-linked document set comprising a plurality of documents, wherein at least a subset of the documents contain a plurality of links to other documents, each document being associated with a host, the method comprising:
- a) selecting a host;
 - b) selecting at random a document associated with the host;
 - c) retrieving the selected document;
 - d) responsive to occurrence of a random event:
 - d.1) selecting at random a host from among the previously selected hosts; and
 - d.2) repeating b) through e) until a predetermined condition is met; and
 - e) responsive to non-occurrence of the random event:
 - e.1) selecting at random a link in the retrieved document;
 - e.2) retrieving a document referenced by the selected link; and
 - e.3) repeating d) and e) until a predetermined condition is met.

28. (Previously presented) A computer-implemented method for measuring relative quality of a target document in a document set comprising a plurality of documents, wherein at least a subset of the documents contain a plurality of links to other documents, the method comprising:

- a) performing a two-level random walk among documents within a document set, by:
 - a.1) initializing a host set;
 - a.2) initializing a document set for each host in the host set;
 - a.3) selecting at random a host from the host set;
 - a.4) responsive to occurrence of a random event:
 - a.4.1) selecting at random a host from among the previously selected hosts;
 - a.5) responsive to non-occurrence of the random event:
 - a.5.1) selecting at random a document from the document set of the selected host; and
 - a.5.2) responsive to the selected document containing at least one link:
 - a.5.2.1) selecting at random a link from the selected document;
 - a.5.2.2) selecting a document corresponding to the selected link;
 - a.5.2.3) selecting a host corresponding to the selected document; and
 - a.5.2.4) adding the selected host to the host set;
 - a.5.2.5) adding the selected document to the document set of the selected host;
 - a.5.2.6) repeating a.5.2.1) through a.5.2.5) until a first predetermined condition is met; and
 - a.6) repeating a.3) through a.5) until a second predetermined condition is met; and

- b) determining a quality metric responsive to the number of documents encountered during the two-level random walk that link to the target document;
- c) determining a quality metric for at least one additional target document; and
- d) ranking the quality metric of the first document with respect to the quality metrics of the additional target documents.

29. (Cancelled).

30. (Previously presented) A computer program product comprising a computer-usable medium having computer-readable code embodied therein for randomly walking through a hypertext-linked document set comprising a plurality of documents, wherein at least a sub-set of the documents contain a plurality of links to other documents, each document being associated with a host, the computer program product comprising:

- a) computer-readable program code devices configured to cause a computer to select a host;
- b) computer-readable program code devices configured to cause a computer to select at random a document associated with the host;
- c) computer-readable program code devices configured to cause a computer to retrieve the selected document;
- d) computer-readable program code devices configured to cause a computer to, responsive to occurrence of a random event:
 - d.1) select at random a host from among the previously selected hosts; and
 - d.2) select at random a document associated with the host; and
 - d.3) retrieve the selected document;
- e) computer-readable program code devices configured to cause a computer to, responsive to non-occurrence of the random event:
 - e.1) select at random a link in the retrieved document; and

- e.2) retrieve a document referenced by the selected link; and
- f) computer-readable program code devices configured to cause a computer to repeat the operations of d) and e) until a predetermined condition is met.

31. (Previously presented) The computer program product of claim 30, wherein the random event comprises:

a generated random number falling within a predetermined range.

32. (Previously presented) The computer program product of claim 30, wherein the document set is the World Wide Web, and wherein each document is a web page.

33. (Canceled).

34. (Canceled).

35. (Previously presented) A computer program product comprising a computer-usable medium having computer-readable code embodied therein for randomly walking through a hypertext-linked document set comprising a plurality of documents, wherein at least a subset of the documents contain a plurality of links to other documents, each document being associated with a host, the computer program product comprising:

- a) computer-readable program code devices configured to cause a computer to initialize a host set;
- b) computer-readable program code devices configured to cause a computer to initialize a document set for each host in the host set;
- c) computer-readable program code devices configured to cause a computer to select at random a host from the host set;

- d) computer-readable program code devices configured to cause a computer to select at random a document from the document set of the selected host;
- e) computer-readable program code devices configured to cause a computer to, responsive to the selected document containing at least one link:
 - e.1) select at random a link from the selected document;
 - e.2) select a document corresponding to the selected link;
 - e.3) select a host corresponding to the selected document; and
 - e.4) add the selected host to the host set;
 - e.5) add the selected document to the document set of the selected host; and
 - e.6) repeat the operations of e.1) through e.5) until a first predetermined condition is met; and
- f) computer-readable program code devices configured to cause a computer to repeat the operations of c) through e) until a second predetermined condition is met.

36. (Original) The computer program product of claim 35, wherein:
the computer-readable program code devices configured to cause a computer to add the selected host to the host set operate responsive to the selected host not being in the host set; and
the computer-readable program code devices configured to cause a computer to add the selected document to the document set of the selected host operate responsive to the selected document not being in the document set of the selected host.

37. (Previously presented) A computer program product comprising a computer-usable medium having computer-readable code embodied therein for randomly walking through a hypertext-linked document set comprising a plurality of documents, wherein at least a sub-set of the documents contain a plurality of

links to other documents, each document being associated with a host, the computer program product comprising:

- a) computer-readable program code devices configured to cause a computer to initialize a host set;
- b) computer-readable program code devices configured to cause a computer to initialize a document set for each host in the host set;
- c) computer-readable program code devices configured to cause a computer to select at random a host from the host set;
- d) computer-readable program code devices configured to cause a computer to select at random a document from the document set of the selected host;
- e) computer-readable program code devices configured to cause a computer to, responsive to non-occurrence of a random event, and further responsive to the selected document containing at least one link:
 - e.1) select at random a link from the selected document;
 - e.2) select a document corresponding to the selected link;
 - e.3) select a host corresponding to the selected document; and
 - e.4) add the selected host to the host set;
 - e.5) add the selected document to the document set of the selected host; and
 - e.6) repeat the operations of e.1 through e.5) until a first predetermined condition is met; and
- f) computer-readable program code devices configured to cause a computer to repeat the operations of c) through e) until a second predetermined condition is met.

38. (Previously presented) The computer program product of claim 37, wherein the random event comprises:

a generated random number falling within a predetermined range.

39. (Original) The computer program product of claim 35, wherein the hypertext-linked document set is the World Wide Web, and wherein each document is a web page.

40. (Original) The computer program product of claim 39, wherein each host corresponds to a domain.

41. (Original) A computer program product comprising a computer-usable medium having computer-readable code embodied therein for measuring relative quality of a search engine index, the computer program product comprising:

- a) computer-readable program code devices configured to cause a computer to perform a two-level random walk among documents within a document set;
- b) computer-readable program code devices configured to cause a computer to, for each document encountered in the random walk, determine whether the document is indexed by the search engine index; and
- c) computer-readable program code devices configured to cause a computer to aggregate the results of the operations of b).

42. (Original) The computer program product of claim 41, wherein at least a subset of the documents contain a plurality of links to other documents, each document being associated with a host, and wherein the computer-readable program code devices configured to cause a computer to perform a two-level random walk comprise:

- a.1) computer-readable program code devices configured to cause a computer to select a host;
- a.2) computer-readable program code devices configured to cause a computer to select at random a document associated with the host;
- a.3) computer-readable program code devices configured to cause a computer to retrieve the selected document;

- a.4) computer-readable program code devices configured to cause a computer to select at random a link in the retrieved document;
- a.5) computer-readable program code devices configured to cause a computer to retrieve a document referenced by the selected link;
and
- a.6) computer-readable program code devices configured to cause a computer to repeat the operations of a.4) and a.5) until a predetermined condition is met.

43. (Canceled).

44. (Previously presented) The computer program product of claim 41, wherein at least a subset of the documents contain a plurality of links to other documents, each document being associated with a host, and wherein the computer-readable program code devices configured to cause a computer to perform a two-level random walk comprise:

- a.1) computer-readable program code devices configured to cause a computer to initialize a host set;
- a.2) computer-readable program code devices configured to cause a computer to initialize a document set for each host in the host set;
- a.3) computer-readable program code devices configured to cause a computer to select at random a host from the host set;
- a.4) computer-readable program code devices configured to cause a computer to select at random a link from a document in the document set of the selected host;
- a.5) computer-readable program code devices configured to cause a computer to add the host referenced by the link to the host set;
- a.6) computer-readable program code devices configured to cause a computer to add the document referenced by the link to the document set of the selected host;

a.7) computer-readable program code devices configured to cause a computer to, responsive to the selected document containing at least one link:

a.7.1) select at random a link from the selected document;

a.7.2) select a document corresponding to the selected link;

a.7.3) select a host corresponding to the selected document;

a.7.4) repeat the operations of a.5) through a.8) until a predetermined condition is met; and

a.8) computer-readable program code devices configured to cause a computer to, responsive to the selected document not containing at least one link, repeat the operations of a.3) through a.8) until a predetermined condition is met.

45. (Original) The computer program product of claim 44, wherein:

the computer-readable program code devices configured to cause a computer to add the selected host to the host set are configured to cause a computer to add the selected host responsive to the selected host not being in the host set; and

the computer-readable program code devices configured to cause a computer to add the selected document to the document set of the selected host are configured to cause a computer to add the selected document responsive to the selected document not being in the document set of the selected host.

46. (Previously presented) The computer program product of claim 41, wherein each document contains a plurality of words, and wherein the computer-readable program code devices configured to cause a computer to determine whether the document is indexed by the search engine index comprise computer-readable program code devices configured to, for each document encountered in the random walk:

b.1) select at least one word from the document;

- b.2) perform a query on the search engine index based on the selected at least one word, to obtain search results; and
- b.3) determine whether the document is included in the obtained search results.

47. (Original) The computer program product of claim 46, wherein the computer-readable program code devices configured to select at least one word from the document comprise computer-readable program code devices configured to select at least one word based on rarity.

48. (Previously presented) A computer program product comprising a computer-usable medium having computer-readable code embodied therein for measuring relative quality of a target document in a document set, the computer program product comprising:

- computer-readable program code devices configured to cause a computer to perform a two-level random walk among documents within a document set; and
- computer-readable program code devices configured to cause a computer to determine a quality metric responsive to the number of times the target document is encountered in the random walk.

49. (Previously presented) A computer program product comprising a computer-usable medium having computer-readable code embodied therein for measuring relative quality of a target document in a document set comprising a plurality of documents, wherein at least a subset of the documents contain a plurality of links to other documents, the computer program product comprising:

- computer-readable program code devices configured to cause a computer to perform a two-level random walk among documents within a document set; and
- computer-readable program code devices configured to cause a computer to determine a quality metric responsive to the number of

documents encountered during the two-level random walk that link to the target document.

50. (Previously presented) The computer program product of claim 49, wherein the computer-readable program code devices configured to cause a computer to determine a quality metric comprise computer-readable program code devices configured to cause a computer to determine a quality metric responsive to the number of documents that link to the target document, and responsive to the quality metric of the linking documents.

51. (Previously presented) The computer program product of claim 49, wherein the computer-readable program code devices configured to cause a computer to determine a quality metric comprise computer-readable program code devices configured to cause a computer to determine a value for:

$$R(p) = d / T + (1 - d) \sum_{i=1}^k R(p_i) / C(p_i)$$

where:

$R(p)$ is the PageRank of target document p ;

$R(p_i)$ is the PageRank of document p_i ;

T is the total number of documents in the document set;

d is a damping factor such that $0 < d < 1$;

documents p_1, \dots, p_k each contain at least one link to target document p ;

and

$C(p_i)$ is the number of links out of document p_i .

52. (Previously presented) A computer program product comprising a computer-usable medium having computer-readable code embodied therein for measuring relative quality of a target document in a document set comprising a plurality of documents, wherein at least a subset of the documents contain a plurality of links to other documents, and wherein each document is associated with a host, the computer program product comprising:

computer-readable program code devices configured to cause a computer to perform a two-level random walk among documents within a document set, by:

- a.1) selecting a host;
- a.2) selecting at random a document associated with the host;
- a.3) retrieving the selected document;
- a.4) responsive to occurrence of a random event:
 - a.4.1) selecting at random a host from among the previously selected hosts; and
 - a.4.2) selecting at random a document associated with the host; and
 - a.4.3) retrieving the selected document;
- a.5) responsive to non-occurrence of the random event;
- a.6) selecting at random a link in the retrieved document; and
- a.7) retrieving a document referenced by the selected link; and
- a.8) repeating the operations of a.4) to a.7) until a predetermined condition is met; and

computer-readable program code devices configured to cause a computer to determine a quality metric responsive to the number of documents encountered during the two-level random walk that link to the target document.

53. (Previously presented) A computer program product comprising a computer-usable medium having computer-readable code embodied therein for measuring relative quality of a target document in a document set comprising a plurality of documents, wherein at least a subset of the documents contain a plurality of links to other documents, wherein each document is associated with a host, the computer program product comprising:

computer-readable program code devices configured to cause a computer to perform a two-level random walk among documents within a document set, by:

- a.1) initializing a host set;
 - a.2) initializing a document set for each host in the host set;
 - a.3) selecting at random a host from the host set;
 - a.4) responsive to occurrence of a random event:
 - a.4.1) selecting at random a host from among the previously selected hosts;
 - a.5) responsive to non-occurrence of the random event:
 - a.5.1) selecting at random a document from the document set of the selected host;
 - a.5.2) adding the selected host to the host set;
 - a.5.3) adding the selected document to the document set of the selected host;
 - a.5.4) responsive to the selected document containing at least one link:
 - a.5.4.1) selecting at random a link from the selected document;
 - a.5.4.2) selecting a document corresponding to the selected link;
 - a.5.4.3) selecting a host corresponding to the selected document; and
 - a.5.4.4) repeating the operations of a.5.2) through a.5.4.3) until a first predetermined condition is met; and
 - a.6) repeating the operations of a.3) through a.5.4.4) until a second predetermined condition is met; and
- computer-readable program code devices configured to cause a computer to determine a quality metric responsive to the number of documents encountered during the two-level random walk that link to the target document.

54. (Previously presented) The computer program product of claim 49, further comprising:

- c) computer-readable program code devices configured to cause a computer to determine a quality metric for at least one additional target document; and
- d) computer-readable program code devices configured to cause a computer to rank the quality metric of the first target document with respect to the quality metrics of the additional target documents.

55. (Previously presented) A computer program product comprising a computer-usable medium having computer-readable code embodied therein for randomly walking through a hypertext-linked document set comprising a plurality of documents, wherein at least a subset of the documents contain a plurality of links to other documents, each document being associated with a host, the computer program product comprising:

- a) computer-readable program code devices configured to cause a computer to select a host;
- b) computer-readable program code devices configured to cause a computer to select at random a document associated with the host;
- c) computer-readable program code devices configured to cause a computer to retrieve the selected document;
- d) computer-readable program code devices configured to cause a computer to, responsive to occurrence of a random event:
 - d.1) select at random a host from among the previously selected hosts; and
 - d.2) repeat the operations of b) through e) until a predetermined condition is met
- e) computer-readable program code devices configured to cause a computer to, responsive to non-occurrence of the random event:
 - e.1) select at random a link in the retrieved document;
 - e.2) retrieve a document referenced by the selected link; and

- e.3) repeat the operations of d) and e) until a predetermined condition is met.

56. (Previously presented) A computer program product comprising a computer-usable medium having computer-readable code embodied therein for measuring relative quality of a target document in a document set comprising a plurality of documents, wherein at least a subset of the documents contain a plurality of links to other documents, the computer program product comprising:

- a) computer-readable program code devices configured to cause a computer to perform a two-level random walk among documents within a document set by:
 - a.1) initializing a host set;
 - a.2) initializing a document set for each host in the host set;
 - a.3) selecting at random a host from the host set;
 - a.4) responsive to occurrence of a random event:
 - a.4.1) selecting at random a host from among the previously selected hosts;
 - a.5) responsive to non-occurrence of the random event:
 - a.5.1) selecting at random a link from a document in the document set of the selected host;
 - a.5.2) adding the host referenced by the link to the host set;
 - a.5.3) adding the document referenced by the link to the document set of the selected host;
 - a.5.4) responsive to the selected document containing at least one link:
 - a.5.4.1) selecting at random a link from the selected document;
 - a.5.4.2) selecting a document corresponding to the selected link;
 - a.5.4.3) selecting a host corresponding to the selected document;

- a.5.4.4) repeating the operations of a.5.2) through a.5.4.3) until a first predetermined condition is met; and
- a.9) responsive to the selected document not containing at least one link, repeat the operations of a.3) through a.5.4.4) until a second predetermined condition is met;
- b) computer-readable program code devices configured to cause a computer to determine a quality metric responsive to the number of documents encountered during the two-level random walk that link to the target document;
- c) computer-readable program code devices configured to cause a computer to determine a quality metric for at least one additional target document; and
- d) computer-readable program code devices configured to cause a computer to rank the quality metric of the first document with respect to the quality metrics of the additional target documents.

57. (Previously presented) A system for randomly walking through a hypertext-linked document set comprising a plurality of documents, wherein at least a subset of the documents contain a plurality of links to other documents, each document being associated with a host, the system comprising:

- a) a host selector;
- b) a random document selector, coupled to the host selector, for selecting at random a document associated with the host;
- c) a document retriever, coupled to the random document selector, for retrieving the selected document; and
- d) a link selector, coupled to the document retriever;

wherein, responsive to occurrence of a random event:

the host selector selects at random a host from among the previously selected hosts;

the random document selector selects at random a document associated with the host; and
the document retriever retrieves the selected document; and
wherein, responsive to non-occurrence of the random event:
the link selector selects at random a link in the retrieved document;
and
the document retriever retrieves a document referenced by the selected link; and
and wherein the link selector, the random document selector, and the document retriever repeat their respective operations until a predetermined condition is met.

58. (Original) A system for measuring relative quality of a search engine index, comprising:

a random walker, for performing a two-level random walk among documents within a document set;
a determination module, coupled to the random walker, for, for each document encountered in the random walk, determining whether the document is indexed by the search engine index; and
a results aggregation module, coupled to the determination module, for aggregating the results of the determination module.

59. (Previously presented) A system for measuring relative quality of a target document in a document set, comprising:

a random walker, for performing a two-level random walk among documents within a document set; and
a determination module, coupled to the random walker, for determining a quality metric responsive to the number of times the target document is encountered in the random walk.

Appl. No. 09/392,170
Appeal Brief dated July 12, 2004
Reply to Office action of April 20, 2004

60. (Previously presented) A system, comprising:
a processor; and
memory containing software executable by the processor;
wherein, by executing the software, the processor initializes a document set, selects an arbitrary hyperlink included in a selected document in the document set, and adds a document referenced by the hyperlink to the document set.
61. (Previously presented) The system of claim 60 wherein the processor further initializes a host set and adds a host referenced by the arbitrary hyperlink to the host set.
62. (Previously presented) The system of claim 60 wherein the processor further determines whether the document referenced by the arbitrary hyperlink is included in a search engine index.

Appl. No. 09/392,170
Appeal Brief dated July 12, 2004
Reply to Office action of April 20, 2004

APPENDIX "B" TO APPEAL BRIEF
"Writing a Web Crawler in the Java Programming Language"

developers.sun.com

>> search tips | Search:

in Dev



The Source for Developers
A Sun Developer Network Site

Products & Technologies
Technical Topics

Developers Home > Products & Technologies > Java Technology > Reference > Technical Articles and
Tips > Developer Technical Articles & Tips > Third-Party Technologies >

Join a Sun Developer Net
Profile and Registratio

Article

Writing a Web Crawler in the Java Programming Language

Printable Page

[Articles Index](#)

By Thom Blum, Doug Keislar, Jim Wheaton, and Erling Wold of Muscle Fish, LLC
January 1998

Everyone uses web crawlers—indirectly, at least! Every time you search the Internet using a service such as Alta Vista, Excite, or Lycos, you're making use of an index that's based on the output of a web crawler. Web crawlers—also known as spiders, robots, or wanderers—are software programs that automatically traverse the Web. Search engines use crawlers to find what's on the Web; then they construct an index of the pages that were found.

However, you might want to use a crawler directly. You might even want to write your own! Here are some possible reasons:

- You want to maintain mirror sites for popular Web sites.
- You need to test web pages and links for valid syntax and structure.
- You want to monitor sites to see when their structure or contents change.
- Your company needs to search for copyright infringements.
- You'd like to build a special-purpose index—for example, one that has some understanding of the content stored in multimedia files on the Web.

This article explains what web crawlers are. It includes a web-crawling demo program, written in the Java programming language, that you can run from your browser. The demo traverses the Web automatically, shows a running list of files it has found, and updates the list each time it finds a new one. You can specify what type of file you want to find. The Java language source code for this demo application is provided as a programming example.

How Web Crawlers Work

Web crawlers start by parsing a specified web page, noting any hypertext links on that page that point to other web pages. They then parse those pages for new links, and so on, recursively. Web crawler software doesn't actually move around to different computers on the Internet, as viruses or intelligent agents do. A crawler resides on a single machine. The crawler simply sends HTTP requests for documents to other machines on the Internet, just as a web browser does when the user clicks on links. All the crawler really does is to automate the process of following links.

Following links isn't greatly useful in itself, of course. The list of linked pages almost always serves some subsequent purpose. The most common use is to build an index for a web search engine, but crawlers are also used for other purposes, such as those mentioned in the previous section.

Muscle Fish uses a crawler to search the Web for audio files. This is a straightforward task, as shown by the demo in the next section. It turns out that searching for audio files is not very different from searching for any other kind of file. On the other hand, *indexing* audio is anything but straightforward. Most search engines, if they handle audio at all, index only textual information that's associated with the sound file. Muscle Fish's approach is to acoustically analyze the audio itself. This feature lets you search for sound files based on how they actually sound—you're not limited to searching for whatever words happen to be located nearby on the same web page. (A forthcoming article and demo program will show this feature.)

A Web-Crawling Demo Program

The simple application shown below crawls the Web, searching for a specified type of file.

Appl. No. 09/392,170
Appeal Brief dated July 12, 2004
Reply to Office action of April 20, 2004

APPENDIX "C" TO APPEAL BRIEF
"Frequently asked Questions about the Mercator Web Crawler"

COMPAQ Copyright © 1999, Compaq Computer Corporation. All rights reserved.

Frequently Asked Questions about the Mercator Web Crawler

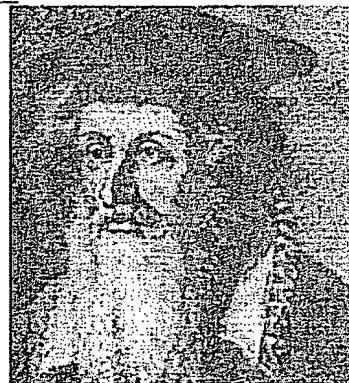
What is a "web crawler"?

A web crawler is a program that systematically fetches web pages. Web crawlers are also referred to as robots, spiders, worms, or wanderers. A web crawler works by fetching a *seed* page, then fetching all the pages this seed page points to, then fetching all the pages *those* pages point to, and so on.

To learn more about web crawlers, refer to Martijn Koster's [web robots pages](#).

Why did you choose the name "Mercator"?

Gerardus Mercator, 1512-1594. Flemish cartographer whose most important innovation was a map, later known as the Mercator projection, on which parallels and meridians are rendered as straight lines spaced so as to produce at any point an accurate ratio of latitude to longitude. Mercator also introduced the term atlas for a collection of maps. -- Encyclopædia Britannica



Our crawler, like the famous cartographer, aims at producing ``maps" of the known (virtual) world that accurately depict its dimensions.

How can I be sure that you will not overload my web server?

Mercator is designed to fetch only one document at a time from any given host. In other words, it will not attempt to fetch multiple documents in parallel. So, Mercator places about the same load on your system that a (speed-reading) human web surfer would.

In practice, Mercator will crawl your site over the course of several hours or days, with short bursts of requests that are close together in time, and long pauses in between.

How can I prevent Mercator from crawling my site?

Mercator observes the [Robots Exclusion Protocol](#). Before attempting to download any document from a site (say www.yoursite.org), Mercator will attempt to download a document with the URL <http://www.yoursite.org/robots.txt>. The `robots.txt` file is created by the web master. It contains a set of rules indicating which parts of a site are off-limits to web crawlers.

Here is a `robots.txt` file that would prevent Mercator from visiting any pages on your site:

```
User-Agent: Mercator
Disallow: /
```


To prevent all web crawlers (not just Mercator) from accessing your site, use the following `robots.txt` file instead:

```
User-Agent: *  
Disallow: /
```

Remember that if Mercator can't access your `robots.txt` file, it has no way of knowing that it should stay out. So, once you have created the file, you should make sure that it is visible by pointing your browser at the URL `http://www.yoursite.org/robots.txt` (replace `www.yoursite.org` by the name of your site).

For the technically interested, Mercator conforms to the latest version of the Robots Exclusion Protocol. In particular, it supports both `Allow` and `Disallow` rules.

I have edited my `robots.txt` file. Why are you *still* crawling my pages?

If Mercator visited your site earlier and did not find a `robots.txt` file, it will not attempt to fetch it again for a week (in accordance with section 3.4 of the latest specification of the Robots Exclusion Protocol). If Mercator did find a `robots.txt` file, it remembers the information contained there until the page expires. The expiration date of the `robots.txt` page was transmitted by your web server. If no expiration date was indicated, Mercator assumes that your `robots.txt` file is valid for one week from the time it was downloaded.

How can I prevent crawling if I can't create a `robots.txt` file?

Many people's web pages are stored on web servers over which they have no administrative control. For example, their ISP may allow them to put personal web pages on the ISP's web server, but it may not allow them to change the web server's `robots.txt`.

To accommodate people in this situation, Mercator also observes the Robots META tag. If you don't want any web crawler to process your pages or to follow the links contained in them, your web pages should start as follows:

```
<HTML>  
<HEAD>  
<META NAME="robots" CONTENT="noindex,nofollow">  
...  
</HEAD>  
...  
</HTML>
```

If you are using a special HTML editor to create your web pages, consult its user manual on how to insert special HTML tags into the pages. You have to add a META tag to the HEAD section of the page. The tag should have two attributes: an attribute `NAME` set to `robots`, and an attribute `CONTENT` set to `noindex,nofollow`. Many HTML editors provide special dialogue boxes for specifying META tags.

How can I report problems to you?

If you have followed our instructions on how to prevent Mercator from crawling your web pages, and it is still fetching your pages, please send us email at mercator@pa.dec.com. Please include in your message:

- The URLs of the pages Mercator has erroneously fetched.
- If possible, the relevant lines of your web server log.
- The method of robots exclusion you are using.
- A valid return address, so we can get back to you.

HP SRC Classic Lab

Mail Stop 1250

1501 Page Mill Road, Palo Alto, CA 94304

Tel: (650) 857-2361 Fax: (650) 852-8186



Send comments to the owner of this page.

Last modified: Tuesday, 23-Mar-1999 10:31:23 PST

[Legal Statement](#) [Privacy Statement](#)